

## AMENDMENT TO CLAIMS

Please amend claims 43, 55 and 68. All pending claims are reproduced below, including those that remain unchanged.

1-42. (Canceled).

43. (Currently Amended) A computer-implemented method for processing XML document, comprising:

parsing an XML document to generate a stream of discrete pieces of the XML document;

accepting [[an]] a discrete piece of the XML document from the stream of discrete pieces and keeping in memory only said discrete piece of the XML document from the stream at any one time;

performing, on one or more microprocessors, a match on said discrete piece of the XML document from the stream and notifying an observer when the discrete piece of the XML document is a matched discrete piece of the XML document, wherein when the discrete piece of the XML document is not a matched discrete piece of the XML document the observer is not notified;

listening for the matched discrete piece of the XML document and passing it to a first user object;

returning said discrete piece of the XML document to the stream; and

pulling said discrete piece of the XML document from the stream for the use of a subsequent user object, wherein the subsequent user object handles said discrete piece of the XML document when there is another match.

44. (Canceled)

45. (Previously Presented) The method according to claim 43, further comprising:

keeping only a portion of the XML document in memory at any given time.

46. (Previously Presented) The method according to claim 43, further comprising:

representing the XML document in a hierarchical structure, which is a XML tree with each node containing a portion of the document.

47. (Previously Presented) The method according to claim 46, wherein:  
the parsing of the XML document comprises the steps of:  
traversing the XML tree and adding visited nodes into a data structure;  
processing the nodes in the data structure and generating a discrete piece of the XML document for each node; and  
appending the discrete piece of the XML document to the output stream.
48. (Previously Presented) The method according to claim 47, wherein:  
the XML tree is traversed using a breath-first or depth-first search.
49. (Previously Presented) The method according to claim 47, wherein:  
the data structure is processed using a first-in-first-out approach.
50. (Previously Presented) The method according to claim 43, further comprising:  
knowing the schema of the XML document and foreseeing the coming events.
51. (Previously Presented) The method according to claim 43, wherein:  
the match is an expression-based match, which uses an XPath query.
52. (Previously Presented) The method according to claim 46, further comprising:  
keeping, cloning and destroying the entirety or a portion of a sub-tree descending from a node in the XML tree.
53. (Previously Presented) The method according to claim 43, further comprising:  
accepting another discrete piece of the XML document at said time.
54. (Previously Presented) The method according to claim 43, further comprising:  
accepting another discrete piece of the XML document at a different time.

55. (Currently Amended) A system implemented using a computer to process XML document, comprising:

- a streaming parser operable to parse an XML document to generate a stream of discrete pieces of the XML document;

- one or more matching components to perform the steps of:

- accepting [[an]] a discrete piece of the XML document from the stream of discrete pieces and keeping in memory only said discrete piece of the XML document from the stream at any one time;

- performing a match on said discrete piece of the XML document from the stream and notifying an observer when the discrete piece of the XML document is a matched discrete piece of the XML document, wherein when the discrete piece of the XML document is not a matched discrete piece of the XML document the observer is not notified;

- listening for the matched discrete piece of the XML document and passing it to a first user object;

- returning said discrete piece of the XML document to the stream; and

- pulling said discrete piece of the XML document from the stream for the use of a subsequent user object, wherein the subsequent user object handles said discrete piece of the XML document when there is another match.

56. (Previously Presented) The system according to claim 55, wherein:  
the XML document is represented in a hierarchical structure.

57. (Previously Presented) The system according to claim 56, wherein:  
the hierarchical structure is a XML tree with each node containing a portion of the document.

58. (Previously Presented) The system according to claim 57, wherein:  
the streaming parser generates the stream of discrete pieces of the XML document by:  
traversing the XML tree and adding visited nodes into a data structure;  
processing the nodes in the data structure and generating a discrete piece of the XML document for each node; and

appending the discrete piece of the XML document to the output stream.

59. (Previously Presented) The system according to claim 58, wherein:  
the XML tree is traversed using a breath-first or depth-first search.
60. (Previously Presented) The system according to claim 58, wherein:  
the data structure is a queue.
61. (Previously Presented) The system according to claim 58, wherein:  
the data structure is processed using a first-in-first-out approach.
62. (Previously Presented) The system according to claim 55, wherein:  
the one or more matching components keep only a portion of the XML document in memory at any given time.
63. (Previously Presented) The system according to claim 55, wherein:  
the one or more matching components know the schema of the XML document and operate to foresee coming discrete pieces of the XML document.
64. (Previously Presented) The system according to claim 55, wherein:  
the match is an expression-based match, which uses an XPath query.
65. (Previously Presented) The system according to claim 55, wherein:  
the one or more matching components keep, clone and destroy the entirety or a portion of a sub-tree descending from a node in the XML tree.
66. (Previously Presented) The system according to claim 55, wherein:  
the one or more matching components perform the step of accepting another discrete piece of the XML document at said time.
67. (Previously Presented) The system according to claim 55, wherein:

the one or more matching components perform the step of accepting another discrete piece of the XML document at a different time.

68. (Currently Amended) A machine readable storage medium having instructions stored thereon that when executed by a processor to:

parse an XML document to generate a stream of discrete pieces of the XML document;

accept [[an]] a discrete piece of the XML document from the stream of discrete pieces and keep in memory only said discrete piece of the XML document from the stream at any one time;

perform a match on said discrete piece of the XML document from the stream and notifying an observer when the discrete piece of the XML document is a matched discrete piece of the XML document, wherein when the discrete piece of the XML document is not a matched discrete piece of the XML document the observer is not notified;

listen for the matched discrete piece of the XML document and passing it to a first user object;

return said discrete piece of the XML document to the stream; and

pull said discrete piece of the XML document from the stream for the use of a subsequent user object, wherein the subsequent user object handles said discrete piece of the XML document when there is another match.

69. (Previously Presented) The machine readable medium of claim 68, further comprising instructions that when executed cause the system to:

keep only a portion of the XML document in memory at any given time.

70. (Previously Presented) The machine readable storage medium of claim 68, further comprising instructions that when executed cause the system to:

know the schema of the XML document and foresee the coming discrete pieces of the XML document.

71. (Previously Presented) The machine readable storage medium of claim 68, further comprising instructions that when executed cause the system to:

represent the XML document in a hierarchical structure, which is a XML tree with each node containing a portion of the document.

72. (Previously Presented) The machine readable storage medium of claim 71, wherein:  
the parsing of the XML document comprises the steps of:  
traversing the XML tree and adding visited nodes into a data structure;  
processing the nodes in the data structure and generating a discrete piece of the XML document for each node; and  
appending the discrete piece of the XML document to the output stream.
73. (Previously Presented) The machine readable storage medium of claim 72, wherein:  
the XML tree is traversed using a breath-first or depth-first search.
74. (Previously Presented) The machine readable storage medium of claim 72, wherein:  
the data structure is processed using a first-in-first-out approach.
75. (Previously Presented) The machine readable storage medium of claim 68, wherein:  
the match is an expression-based match, which uses an XPath query.
76. (Previously Presented) The machine readable storage medium of claim 68, further comprising instructions that when executed cause the system to:  
keep, clone and destroy the entirety or a portion of the sub-tree descending from a node in the tree.
77. (Previously Presented) The machine readable storage medium of claim 68, further comprising instructions that when executed cause the system to:  
accept another discrete piece of the XML document at said time.
78. (Previously Presented) The machine readable storage medium of claim 68, further comprising instructions that when executed cause the system to:  
accept another discrete piece of the XML document at a different time.